CentraleSupélec

Algorithmics and complexity

TD 5/7 – Complexity theory

This tutorial session is about complexity and polynomial reduction.

Exercice 1 : Vertex Cover Problem

We know the Independent Set problem. We are also aware that it is NP-complete.

Instance:

- undirected connected graph $G = (V, E)$,
- $k \in \mathbb{N}^+$

Question : Is there an independent set S (there are no edges between S vertices) such that $|S| \geq k$?

But, we do not know yet the problem of k-Vertex Cover which definition is:

Instance:

- undirected connected graph $G = (V, E)$,
- $k \in \mathbb{N}^+$.

Question : Is there a subset $V' \subseteq V$ of size $|V'| \leq k$, such that any edge $\{u, v\} \in E$ has at least one of its endpoints in V', $u \in V' \vee v \in V'$?

Question 1

Prove that

$$S \text{ is an independent set} \iff V - S \text{ is a vertex cover.}$$

Solution elements :

$\Rightarrow$
Suppose that S is a stable and $e = (u, v)$ an edge. At most one end of (u, v) can be in S. So at least one is in $V - S$. So $V - S$ is a vertex cover.

$\Leftarrow$
Suppose that $V - S$ is a vertex cover and $u \in S, v \in S$. There can be no edge $(u, v) \in E$ (otherwise it would not be covered by $V - S$. So S is a stable.

Question 2

Prove that the k-Vertex Cover problem is NP-complete.

Solution elements :

Step 1. First, we need to show that the problem is NP: we can write an algorithm in Python that checks that a subset V' is a solution to an instance $(G = (V, E), k)$ of the vertex cover problem:

```
def verifier(V,E,k,V'):
  if len(V') > k:
    return False
  for v in V':
    if not v in V:
      return False
  for (u, v) in E:
    if (u not in V') and (v not in V'):
      return False
  return True
```

The verification of V' consists in browsing all the edges by checking if at least one of the two endpoints is in V'. This is done in $\mathcal{O}(|E| \times |V'|)$ if we use a list for V' and we can do better on average if we use a *hashSet* for V'. It doesn't matter, as long as the complexity is polynomial in instance size.

Step 2. We will now show that the vertex cover problem is NP-hard using a polynomial reduction from the Stable problem.

Let $\mathcal{I}_{Stable} = \langle G = (V, E), k\rangle$ be an instance of the k-Stable problem. We simply build an instance $\mathcal{I}_{VC} = \langle G = (V, E), k'\rangle$ of the Vertex-Cover problem: the graph does not change and $k' = |V| - k$. This transformation algorithm is clearly **polynomial**!

We must show that: $\mathcal{I}_{Stable}$ is a positive instance $\iff$ $\mathcal{I}_{VC}$ is a positive instance

Let us suppose that $\mathcal{I}_{Stable}$ is a positive instance of Stable, it admits a solution $S \subseteq V$ with $|S| \geq k$. From the previous question, $V' = V - S$ is a vertex cover and we have $|V'| = |V| - |S| \leq |V| - k = k'$. So $\mathcal{I}_{VC}$ is a positive instance of Vertex Cover which admits V' as solution.

Conversely, if $\mathcal{I}_{VC}$ is a positive instance of Vertex Cover that admits V' ($V' \subseteq V \wedge |V'| \leq k'$) as a solution. From the previous question, $S = V - V'$ is a Stable and we have $|S| = |V| - |V'| \geq |V| - k' = k$. So $\mathcal{I}_{Stable}$ is a positive instance of Stable which admits S as a solution.

Step 3. In conclusion, there is a polynomial reduction from Stable (which is NP-complete and therefore NP-hard) to Vertex Cover, so Vertex Cover is NP-hard. And since it is NP, it is also NP-complete.

Exercise 2 : Set Cover problem

We say that an element e is covered by a set U if e belongs to U. Let U be a finite set and $S = \{S_i, i \in I\} \subset \mathscr{P}(U)$ be a family of subsets of U. The problem consists in finding a cover of all the elements of U with a subfamily of S.

For example, consider $U = \{0, 1, 2, 3, 4\}$ and $S = \{\underbrace{\{0, 1\}}_{S_0}, \underbrace{\{2, 3\}}_{S_1}, \underbrace{\{3, 4\}}_{S_2}, \underbrace{\{0, 1, 2\}}_{S_3}\}$. One can cover U with $\{S_0, S_1, S_2\}$, but the cover composed of the minimal number of subsets is $\{S_2, S_3\}$.

Question 1

Formalize the corresponding decision problem.

Solution elements :

Data :

- a set of elements U
- a family $S \subset \mathscr{P}(U)$ of subsets of U.
- $k \in \mathbb{N}^+$, a positive natural number

Question : Is there a subfamily $S' \subseteq S$ such that:

- S' covers U, i.e.: $U = \cup_{S_i \in S'} S_i$
- the subfamily S' is of size less than or equal to k, i.e.: $card(S') \leq k$.

Question 2

Show that the problem belongs to the complexity class NP.

Solution elements :

We will write in Python a function that returns true if and only if S' is a coverage of U of size less than or equal to k.

We propose a representation of the subfamily $S' \subseteq S$ by a simple list S' of $m = |S|$ booleans $[s'_0, \ldots, s'_{m-1}]$ such that $s'_j = 1$ if and only if $S_j \in S'$ (i.e., the set S_j is retained for the coverage S').

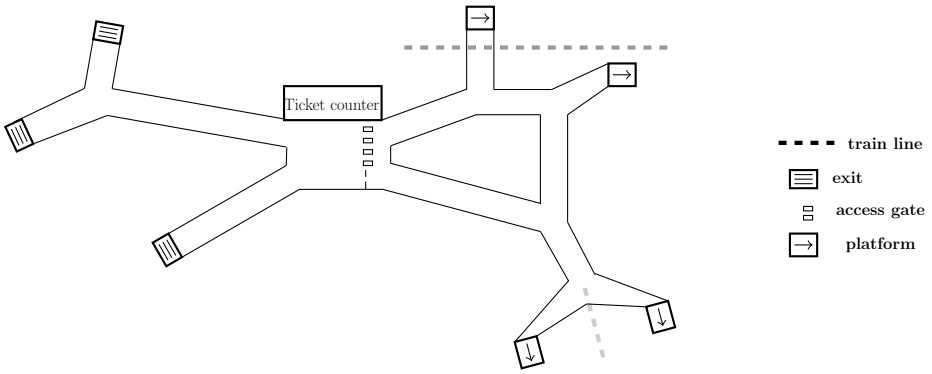
```
U = [0,1,2,3,4]
S = [[0,1],[2,3],[3,4],[0,1,2]]

def SetCover_verif(U, S, SS, k):
    # subfamily size verification
    if sum(SS) > k:
        return False

    # coverage verification
    for i in range(len(U)):
        for j in range(len(S)):
            if (SS[j] == 1) and i in S[j]:
                break
        else:
            return False
    return True


SetCover_verif(U, S, [1, 1, 1, 0], 3) #  >>> True
SetCover_verif(U, S, [1, 0, 1, 0], 2) #  >>> False
SetCover_verif(U, S, [0, 0, 1, 1], 2) #  >>> True
```

We have a double loop $|U| \times |S|$ with each time a search in $\mathcal{O}(|U|)$ (`i in S[j]`), we obtain a polynomial complexity in $O(|U|^2|S|)$. So we can check a solution in polynomial time: the problem is in NP.

Cameras in the subway

In a subway station, 360° cameras can be installed at each passage intersection to control users traffic. To diminish costs, we want to install the minimum number of cameras but each corridor must be monitored by at least one camera.

We consider the following plan of the subway station:

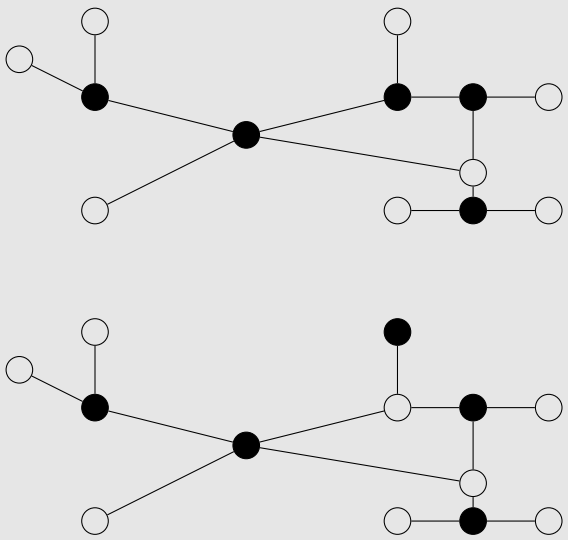


Question 3

Model the subway station using a graph whose nodes represent the intersections, exits or access to the platforms and the edges represent the corridors. Can you cover the entire station with only 8 cameras? And with only 5?

Solution elements :

With 8, there's plenty of solutions! But there is only two solutions with 5 cameras:



Question 4

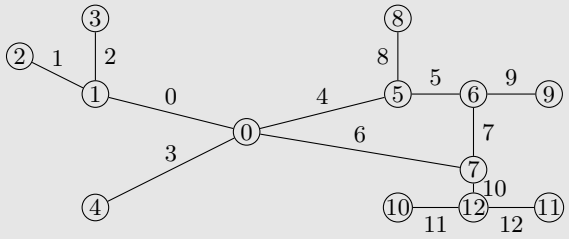
It is natural to model the problem of placing cameras in the subway station as an instance of Vertex Cover seen in the first exercise. We consider the graph $G = (V, E)$ built for the subway station and we want to know whether k vertices (where we will position the cameras) can cover all the edges (corridors).

Alternatively, model this problem as an instance of Set Cover seen in the second exercise.

Hint : Number all corridors

Solution elements :

The corridors are numbered as follows:



We consider the set $U = \{0, \ldots, 12\}$ corresponding to the numbers of edges (of corridors) and the family of subsets: $S_0 = \{0, 3, 4, 6\}$, $S_1 = \{0, 1, 2\}$, $S_2 = \{1\}$, $\ldots$, $S_6 = \{5, 7, 9\}$, $\ldots$, corresponding to each vertex (intersection, exit or platform) and containing the ingoing edges.

the question is: is there a subfamily $S' \subseteq S$ such that $|S'| \leq k$ and S' is covering U?

Question 5

Using a generalisation of the modeling exercise you did on the subway station instance, prove that the Set Cover problem is NP-hard. What can you deduce from this?

Solution elements :

To show NP-hard, we need to propose a polynomial reduction from Vertex Cover to Set Cover.

Given an instance of vertex cover $\mathcal{I}_{VC} = \langle G = (V, E), k \rangle$, we will construct an instance of the Set Cover problem $\mathcal{I}_{SC} = (U, S = (S_i), k)$ such that:

- $U = E$, the elements are the edges of the graph
- The vertices V are numbered from 1 to n ($I = V = [1, n]$). Then we define n subsets S_i of U such that S_i is the set of edges for which the vertex i is an endpoint, $S_i = \{(u, v) \in E | u = i \vee v = i\}$.

This transformation is clearly polynomial in size of the instance $\mathcal{O}(|V| + |E|)$.

Then we have to show that:

$\Rightarrow$ If $V' \subseteq V$ is a vertex cover solution of size at most k then $C = \{S_i, i \in V'\}$ is a family of size less than or equal to k. And it is indeed a set cover, because any element $e = (u, v) \in U = E$, being an edge of G is necessarily covered by at least one vertex, either u, or v, or both. Suppose that u is a covering vertex, we obtain by construction that $e \in S_u \in C$.

$\Leftarrow$ If $C = \{S_i, i \in I'\}$ is a set cover of size less than or equal to k, the set of vertices $V' = I'$ whose numbers are indexing S_i from C is of size at most k. The set V' is a vertex cover in the graph: if there exists an edge $(u, v) = e \in E = U$, we know by construction of C that $S_u \in C$ or $S_v \in C$ because e is covered by C. e is thus covered in the graph either by $u \in V'$, or by $v \in V'$, or by these two vertices.

Conclusion: Set Cover is NP and also NP-hard, so it is NP-complete.